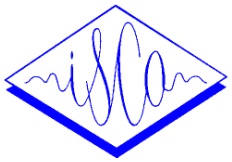

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

SPEAKER VERIFICATION BY INTEGRATING DYNAMIC AND STATIC FEATURES USING SUBSPACE METHOD

M.Nishida and Y.Ariki

Department of Electronics and Informatics
Ryukoku University, Seta, Otsu-shi, Shiga, 520-2194, Japan
nishida@arikilab.elec.ryukoku.ac.jp

ABSTRACT

In speaker recognition, it is a problem that variation of speech features is caused by sentences and time difference. Speech data includes a phonetic information and a speaker information. If they are separated each other, robust speaker verification will be realized by using only the speaker information. However, it is difficult to separate the speaker information from the phonetic information included in speech data at present. From this viewpoint, we propose a speaker verification method using a subspace method based on principal component analysis in order to extract only the speaker information included in speech data. We also propose dynamic and static features of each speaker presented in the speaker eigenspace as well as their integration for robust normalization of speech feature variations. We carried out comparative experiments between the proposed method and conventional GMM to show an effectiveness of our proposed method. As a result, integrated dynamic and static features in speaker eigenspace were shown to be effective for speaker verification.

1. INTRODUCTION

In speaker recognition, it is a problem that speech feature varies depending on sentences and time difference. There have been many studies to suppress the speech feature variations. Typical studies are a feature normalization method and a likelihood normalization method. In the feature normalization method, CMN(Cepstrum Mean Normalization) [1][2] is well known to be effective. In the likelihood normalization method, a likelihood ratio [3] and a posteriori probability [4] are proposed to be effective.

Speech data includes both of phonetic information and speaker information. If they are separated each other, robust speaker verification will be realized by using only the speaker information. However, it is difficult to separate the phonetic information from the speaker information included in speech data at present.

In speaker verification, a GMM(Gaussian Mixture Model) [5] has been conventionally used and is statistically constructed from features of speech data. The GMM is a statistical method and is not a method based on separation of the speaker information from the phonetic information included in speech data.

We have already proposed a speaker identification method using a subspace method [6] based on PCA(Principal Component Analysis) in order to extract only the speaker information included in speech data [7][8]. In this study, we propose a speaker verification method using the subspace method based on PCA. We call an individual speaker subspace constructed by PCA as "speaker eigenspace", because it can be present the speaker information.

Further more, in this study, we propose dynamic features and static features of each speaker presented in the speaker eigenspace as well as their integration for robust speaker verification. The static feature is an averaged speech vector computed for one speech sentence. On the other hand, the dynamic feature is the residual vector between input feature vector and the static feature. Both features are effective to verify the speakers.

In this paper, the results are shown through comparative experiments of the proposed method, a method based on projection distance to the speaker eigenspace and conventional GMM to show an effectiveness of our proposed method.

2. SPEAKER VERIFICATION IN SPEAKER EIGENSPACE

2.1. Speaker Eigenspace Construction

We observe a sequence of training speech data $\{x_t^{(s)}\}$ $(t = 1, 2, \cdots, N)$ of a speaker s in an n-dimensional observation space. The speech data is a sequence of spectral feature vectors $\{x_t^{(s)}\}$ obtained by short time spectral analysis. We compute a mean vector $\mu^{(s)}$ and a covariance matrix $R^{(s)}$ from the training speech data. By eigenvalue decomposition, the covariance matrix $R^{(s)}$ is decomposed as

$$R^{(s)} = \Phi^{(s)} \Sigma^{(s)} {\Phi^{(s)}}^T \tag{1}$$

Here $\Sigma^{(s)}$ is a diagonal matrix whose diagonal components are eigenvalues $\lambda_i^{(s)} (i = 1, \cdots, k, \cdots, n)$ of $R^{(s)}$. $\Phi^{(s)}$ is a matrix whose columns are eigenvectors $\{\varphi_i^{(s)}\} (i = 1, \cdots, k, \cdots, n)$ of $R^{(s)}$.

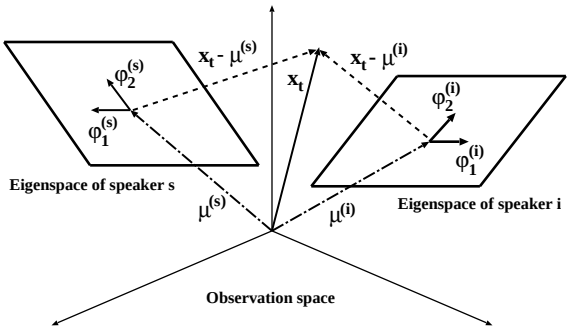

Figure 1: Speaker eigenspace

As shown in Fig.1, eigenvectors $\{\varphi_i^{(s)}\}$ are orthonormal bases and construct an eigenspace(speaker eigenspace) which represents the speech data of speaker s. Therefore, $\Phi^{(s)}$ can be considered to represent the speaker information.

The eigenvectors $\{\varphi_i^{(s)}\}$ for the large eigenvalues up to k numbers construct a k-dimensional speaker eigenspace.

2.2. Verification in Speaker Eigenspace

In verification, we compute a mean distance between input feature vectors $\{x_t\}$ and a speaker eigenspace of a claimed speaker c by computing a projection distance shown in Eq.(2).

$$PD^{(c)} = \frac{1}{N}\sum_{t=1}^{N}\{||x_t - \mu^{(c)}||^2 - \sum_{i=1}^{k}(x_t - \mu^{(c)}, {\varphi_i}^{(c)})^2\} \quad (2)$$

Here N is a total number of frames of input speech data.

Fig.2 shows an example of the projection distance in the 3 -dimensional observation space. The projection distance of speaker c, computed by Eq.(2), is defined by subtracting a square norm of the projection vector(shown by a dotted line in Fig.2) to the speaker c eigenspace(space constructed by ${\varphi_1}^{(c)}$ and ${\varphi_2}^{(c)}$) from a square norm of a residual vector between input feature vector x_t and a mean vector $\mu^{(c)}$ of the training speech data of the speaker c.

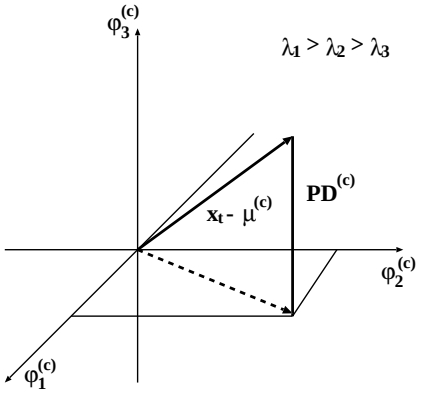

Figure 2: Projection distance

In this study, projection distance $PD^{(c)}$ of speaker c is normalized by the minimum projection distance to other speaker eigenspaces except the claimed speaker as shown in Eq.(3) for robust speaker verification.

$$\hat{PD^{(c)}} = \frac{PD^{(c)}}{\min_{i \neq c} PD^{(i)}} \quad (3)$$

This idea comes from the normalization by likelihood ratio commonly used in the speaker verification in the probabilistic domain.

If the normalized distance is smaller than some threshold, the speaker is accepted as the true person.

3. INTEGRATION OF SPEAKER DYNAMIC AND STATIC FEATURES

3.1. Speaker Dynamic and Static Features

In this study, we define speaker dynamic features and static features in the speaker eigenspace in order to normalize the speech feature variations and propose a speaker verification method by integrating these two features. At first, we describe the speaker dynamic features and static features.

CMN computes a mean vector for a section of input speech data and subtracts the mean vector from the feature vector at each time. "Speaker dynamic feature" is defined as a residual vector produced by subtracting the mean vector of the input speech from the feature vector at each frame. The feature vectors we employed are cepstral coefficients so that the speaker dynamic feature can be computed by CMN of the input feature vectors. The speaker dynamic feature represents a dynamic speaker information and the distance between the speaker dynamic feature of the input speech and the speaker eigenspace is called "dynamic feature distance".

On the other hand, "speaker static feature" is defined as the mean vector itself of the input speech. The speaker static feature represents a static speaker information of input speech and varies time by time even the same sentence is spoken. Therefore a residual vector between the speaker static feature of the input speech and a mean vector of the training speech data can be regarded to present the time difference. The length of the residual vector is called "static feature distance". Fig.3 shows the speaker dynamic features and static features.

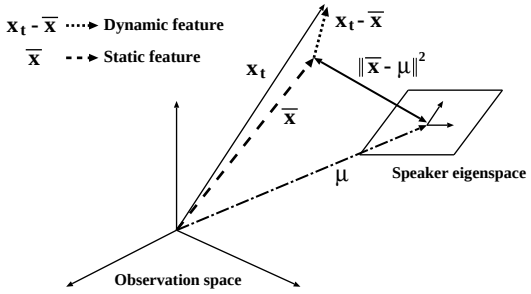

Figure 3: Speaker dynamic and static features

In Fig.3, x_t denotes an input feature vector, $\bar{x}$ denotes a mean vector for a section of input speech and μ denotes a mean vector of speaker training data. The residual vector $x_t - \bar{x}$ between the input feature vector and the mean vector for a section of input speech denotes the speaker dynamic feature. The mean vector $\bar{x}$ for a section of input speech denotes the speaker static feature. A distance $||\bar{x} - \mu||^2$ between the static feature and the mean vector of speaker training data denotes a speaker static feature distance.

Eq.(4) shows the speaker dynamic feature distance $Dist_{dynamic}$ and Eq.(5) shows the speaker static feature distance $Dist_{static}$.

$$Dist_{dynamic} = \frac{1}{N}\sum_{t=1}^{N}\{||x_t - \bar{x}||^2 - \sum_{i=1}^{k}(x_t - \bar{x}, \varphi_i)^2\} \quad (4)$$

$$Dist_{static} = ||\bar{x} - \mu||^2 \quad (5)$$

Here N is a total number of frames of input speech. Eq.(4) denotes the distance of dynamic features to speaker eigenspace after suppressing a variation of time difference. Eq.(5) denotes a distance between the speaker static feature and a mean vector of respective speaker training data.

If the dynamic feature distance is only employed, the time difference presented by the speaker static feature is suppressed. Therefore, it is possible to obtain the speaker verification robustness by integrating the dynamic and static feature distance.

3.2. Integration of Speaker Dynamic and Static Features

Next, we describe a speaker verification method by integrating speaker dynamic features and static features.

A log function draws a gradual curve if the argument value is large so that it can control a variation by the log function. Therefore, it becomes possible to determine a stable threshold by incorporating log distance.

Eq.(6) shows an integrated distance $Dist_{integrated}$ by weighted linear sum of a dynamic feature log distance and a static feature log distance.

$$Dist_{integrated} = \log Dist_{dynamic} + \alpha \log Dist_{static} \quad (6)$$

Since a variance ${\sigma^2}_{static}$ of the static feature log distance $\log Dist_{static}$ is large in comparison with a variance ${\sigma^2}_{dynamic}$ of the dynamic feature log distance $\log Dist_{dynamic}$, an integrated distance $Dist_{integrated}$ of the dynamic and static features is mainly influenced by the static feature distance which represents the time difference. To solve this problem, the weight α is determined by Eq.(7) to balance both of them.

$$\alpha = \frac{{\sigma^2}_{dynamic}}{{\sigma^2}_{static}} \quad (7)$$

In this way, the weight α in Eq.(6) integrates the dynamic speaker information and the static speaker information, and also suppresses the time difference included in the static feature distance.

Eq.(8) shows a normalization method for the integrated distance of the dynamic feature distance and the static feature distance.

$$\hat{Dist}^{(c)}_{integrated} = Dist^{(c)}_{integrated} / \min_{i \neq c} Dist^{(i)}_{integrated} \quad (8)$$

In this normalization method, the integrated distance $Dist^{(c)}_{integrated}$ to the claimed speaker eigenspace is divided by a minimum distance to other speaker eigenspaces except the claimed speaker.

4. SPEAKER VERIFICATION EXPERIMENTS

4.1. Comparative Experiments with Speaker Eigenspace

In the experiment, sentences were uttered by 15 speakers(10 males and 5 females) at three time sessions over ten months. The analytical condition of speech data is shown in Table 1. The speech data was sampled at 12kHz and was parameterized using 16 LPC cepstrum. The analysis window size was 20ms with 5ms overlap.

The 15 speakers are divided into two; One is a true speaker(customer) and the other is 14 impostors for each speaker. Therefore, the total numbers of customers and impostors were 15 and 14 respectively. 10 sentences uttered at first time session were used for training. Each 15 sentences uttered at second and third time sessions were used for evaluation.

A weight α in Eq.(6) for the integrated distance of dynamic and static features was determined as α = 0.06 by computing according to Eq.(7) using the training data.

We carried out comparative experiments between the proposed method and the projection distance to the speaker eigenspace to show an effectiveness of our proposed method.

Table 1: Analytical condition

Sampling frequency	12kHz
Frame length	20ms
Frame period	5ms
Window type	Hamming window
Features	LPC Cepstrum(16 orders)

In each method, verification results for the test data uttered at second and third time sessions are shown in Fig.4 and Table 2.

Fig.4 shows an EER(Equal Error Rate) as a function of speaker eigenspace dimension changing from 1 to 16. In Fig.4, "PCA" denotes a result for the projection distance to the speaker eigenspace, "PCA+CMN" denotes a result when we carried out CMN for the test data in the projection distance to speaker eigenspace(only the dynamic feature) and "Dynamic and static" denotes a result by the integration of dynamic and static features.

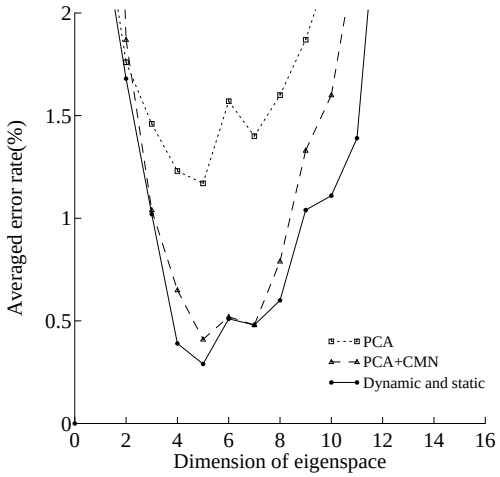

Figure 4: Comparative result in speaker eigenspace (1)

As a result shown in Fig.4, CMN was effective as the normalization of the speech feature variations. The integration of dynamic and static features was effective. As a result of experiments, the optimal dimension of the speaker eigenspace was 5 and in this case, we could construct the speaker eigenspace which represents the best speaker information. If the dimension increases, the eigenspaces of speakers begin to overlap and verification performance decreases.

Table 2: Comparative result in speaker eigenspace (2)

	EER(%)
PCA (5 dimension)	1.17
PCA (5 dimension) + CMN	0.41
Dynamic and static (5 dimension, α=0.06)	0.29

Table 2 shows the EER when the verification performance was the best. As a result from Table 2, the EER was 1.17% by the projection distance

to the speaker eigenspace and the EER was 0.41% by the CMN for the test data, so that the EER was reduced by 65% by the CMN. On the other hand, the EER was 0.29% by integrating dynamic and static features in the speaker eigenspace, so that the EER was reduced by 75% by the proposed method.

As a result from Table 2, the integration method of dynamic and static features in the speaker eigenspace was shown to be robust for the speech feature variations.

4.2. Comparative Experiments with GMM

For the same experimental data under the same analytical condition shown in section4.1., the speaker verification experiments were carried out using GMM with 64 mixture densities and diagonal covariance matrices. In the experiments, we used a Sun Ultra30 (CPU:248MHz,Memory:128MB).

Comparative experiments were carried out between the integration method of dynamic and static features in speaker eigenspace and conventional GMM to show an effectiveness of our proposed method. We carried out the experiments by two methods; with or without CMN for both training and test data. The experimental result is shown in Table 3. The normalization was carried out based on the likelihood ratio shown in Eq.(9).

$$\log \hat{P}(X|S_c) = \log P(X|S_c) - \max_{i \neq c} \log P(X|S_i) \quad (9)$$

In this normalization method, a maximum log likelihood of the other speakers except the claimed speaker is subtracted from a log likelihood $\log P(X|S_c)$ of the claimed speaker c.

Table 3: Comparative result of GMM

	EER(%)
GMM (64md)	0.68
GMM (64md) + CMN	0.34
Dynamic and static (5 dimension, α=0.06)	0.29

The EER by GMM(64 mixture densities) without CMN and the EER with the CMN for both training and test data are shown in Table 3. As a result from Table 3, the GMM showed 0.68% EER when the CMN was not carried out for both training and test data. The GMM showed 0.34% EER when the CMN was carried out for both training and test data so that the EER was reduced by 50% by the CMN. The proposed method showed 0.29% EER which is less than the conventional GMM.

4.3. Processing time

We investigated the processing time and template size for the integration method of dynamic and static features in speaker eigenspace(5 dimension), comparing with GMM(1.0). The training time is a time required for training a customer by 10 sentences(4 sec at average for a sentence). The verification time is a time required for verifying a sentence. The template size is a required memory size for a model of a customer.

As a result from Table 4, the training time of the integration method of dynamic and static features in speaker eigenspace was 0.0056 compared with GMM(1.0) and the verification time was 0.25 in terms of ratio. The template size was 0.03 compared with GMM(1.0) in terms of ratio. Therefore, the computation time and memory of the proposed method are extremely less than that of GMM. In the integration method of dynamic and static features in speaker eigenspace, the exact training time was about 1.5sec and the verification time was about 0.1sec(the template size was about 1.0KB) so that real time processing is feasible by the proposed method.

Table 4: Processing time and template size

	training time	verification time	template size
GMM (64md)	1.0	1.0	1.0
Dynamic and static (5 dimension)	0.0056	0.25	0.03

5. CONCLUSION

In this study, we proposed a speaker verification method using the subspace method and constructed the speaker eigenspace based on PCA in order to extract only the speaker information included in speech data. Moreover, we proposed dynamic and static features of each speaker presented in the speaker eigenspace as well as their integration for robust normalization of speech feature variations.

As a result of the speaker verification experiments, the EER was reduced by 75% by the proposed method compared with the projection distance to the speaker eigenspace. The verification performance of the proposed method was almost same as the conventionally used GMM. The computation time and memory of the proposed method are extremely less than GMM so that real time processing is feasible by the proposed method. From these results, the integration method of dynamic and static features in speaker eigenspace which we proposed in this study was shown to be robust for speech feature variations and was shown to be effective for speaker verification.

Future works will be the improvement of a verification performance by constructing a discriminant function based on correlation to the other classes.

6. REFERENCES

1. S.Furui, “Cepstral Analysis Technique for Automatic Speaker Verification,” IEEE Trans.ASSP, vol.29, no.2, pp.254-272, 1981.
2. A.E.Rosenberg, C-H.Lee and F.K.Soong, “Cepstral Channel Normalization Techniques for HMM-Based Speaker Verification,” Proc.ICSLP, vol.4, pp.1835-1838, 1994.
3. A.E.Rosenberg, J.Delong, C-H.Lee, B-H.Juang and F.K.Soong, “The Use of Cohort Normalized Scores for Speaker Verification,” Proc.ICSLP, vol.1, pp.599-602, 1992.
4. T.Matsui and S.Furui, “Speaker Adaptation of Tied-Mixture-Based Phoneme Models for Text- Prompted Speaker Recognition,” Proc.ICASSP, vol.I, pp.125-128, 1994.
5. M.El-Maliki and A.Drygajlo, “Missing Features Detection and Handling for Robust Speaker Verification,” Proc.EUROSPEECH, pp.975-978, 1999.
6. E.Oja,“ Subspace Methods of Pattern Recognition,” Research Studies Press, England, 1983.
7. Y.Ariki and K.Doi, “Speaker Recognition based on Subspace Method,” ICSLP94, pp.1859-1862, 1994.
8. Y.Ariki, S.Tagashira and M.Nishijima, “Speaker Recognition and Speaker Normalization by Projection to Speaker Subspace,” ICASSP96, sp9.1, pp.319-322, 1996.